

Министерство образования и науки Украины  
Харьковский национальный университет имени В.Н.Каразина

**Л.Н.Попова**

## **КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ЗАДАЧ НЕБЕСНОЙ МЕХАНИКИ**

Методическое пособие  
по лабораторному практикуму «Компьютерное моделирование в механике»

Харьков – 2009

УДК 004.942:521.1(075.8)

ББК 32.973я73+22.62я73

П57

*Утверждено ученым советом механико-математического факультета  
Харьковского национального университета имени В.Н.Каразина  
(протокол № 3 от 20 марта 2009 года)*

**Рецензенты:** доктор физико-математических наук, профессор

**Пацегон Николай Федорович,**

профессор кафедры теоретической механики

Харьковского национального университета имени В.Н.Каразина

кандидат физико-математических наук, доцент

**Игнатович Светлана Юрьевна,**

доцент кафедры дифференциальных уравнений и управления

Харьковского национального университета имени В.Н.Каразина

Методическое пособие содержит указания к лабораторному практикуму по компьютерному моделированию в механике. Рассматриваются три задачи небесной механики, связанные с невозмущенным движением планет Солнечной системы, переводом космического аппарата с одной круговой орбиты на другую и движением тела в поле тяготения планеты. Описываются математическая формулировка задач, методы решения и порядок выполнения лабораторных работ по созданию их компьютерных моделей в системе Delphi. Пособие рекомендовано студентам третьего курса механико-математического факультета, обучающимся по специальности «механика», а также студентам других специальностей физико-математического профиля.

УДК 004.942:521.1(075.8)

ББК 32.973я73+22.62я73

© ХНУ имени В.Н.Каразина, 2009

© Попова Л.Н., 2009

© Макет обложки Дончик И.Н., 2009

## ПРЕДИСЛОВИЕ

Настоящее пособие содержит указания к выполнению заданий по компьютерному моделированию задач небесной механики. Оно рассчитано на студентов третьего курса механико-математического факультета, обучающихся по специальности «механика», и использует знания, полученные ими при изучении основных математических дисциплин, теоретической механики и информатики. Задания выполняются студентами в ходе лабораторного практикума по компьютерному моделированию в механике, который проводится в компьютерном классе в течение одного семестра. Цель практикума – привить студентам навыки самостоятельной работы по созданию программного обеспечения современных задач механики.

В пособии представлены проекты для среды визуального программирования Delphi, связанные с тремя задачами небесной механики: движение планет Солнечной системы (проект SOLAR), перевод космического аппарата с одной круговой орбиты на другую (проект HOHMANN) и движение тела в поле тяготения планеты (проект GRAVITY). Подробно описана последовательность работы над первым проектом, для двух других приводятся основные расчетные формулы, общий сценарий работы проектов и примерный план лабораторных занятий по их реализации. Проект HOHMANN был включен в программу лабораторного практикума в 2006-2007 учебном году, проект GRAVITY – в 2007-2008 учебном году, а проект SOLAR разрабатывался студентами в пятом семестре 2008-2009 учебного года. Работа над проектами завершалась выполнением дополнительного индивидуального задания, которое получал каждый студент перед зачетом. Перечень заданий по программе SOLAR и краткие указания к ним содержатся в конце первого раздела.

В конце приводится список литературы по небесной механике и программированию, которая может быть полезна студентам, желающим подробнее познакомиться с рассматриваемыми задачами и методами визуального моделирования. Вошедшие в список книги имеются в фонде Центральной научной библиотеки университета и доступны студентам.

## ПРОЕКТ SOLAR

### 1.1. Невозмущенное движение планет

В Солнечную систему входит восемь планет: Меркурий, Венера, Земля, Марс, Юпитер, Сатурн, Уран и Нептун. До 24 августа 2006 года Плутон считался девятой планетой Солнечной системы, но был лишен этого статуса решением XXVI Генеральной ассамблеи Международного астрономического союза, так как не удовлетворяет новому определению термина «планета», принятому ассамблеей. Теперь Плутон относится к карликовым планетам Солнечной системы.

Программа SOLAR представляет на экране ПК невозмущенное движение планет Солнечной системы. Движение рассматривается относительно гелиоцентрической эклиптической системы координат. Начало  $S$  этой системы находится в центре Солнца, плоскость  $Sxy$  совпадает с плоскостью эклиптики, ось  $x$  проходит через точку весеннего равноденствия  $\gamma$ , а ось  $z$  - через северный полюс эклиптики  $\Pi$  (рис.1).

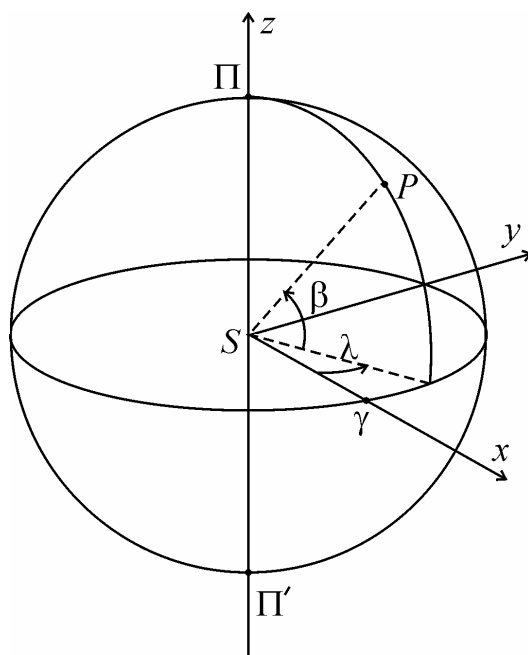


Рис. 1. Гелиоцентрическая эклиптическая система координат

Движение планет описывается решением задачи двух тел и подчиняется законам Кеплера:

1. Все планеты описывают вокруг Солнца плоские кривые, следуя закону площадей.
2. Орбиты планет суть эллипсы, в одном из фокусов которых находится Солнце.
3. Квадраты периодов обращения планет вокруг Солнца пропорциональны кубам больших полуосей их орбит.





$$\begin{aligned}
 j_x &= -\cos\Omega \sin\omega - \sin\Omega \cos\omega \cos i \\
 j_y &= -\sin\Omega \sin\omega + \cos\Omega \cos\omega \cos i \\
 j_z &= \cos\omega \sin i.
 \end{aligned}
 \tag{3}$$

Положение планеты в системе  $Sxyz$  можно определить также сферическими координатами: расстоянием до Солнца  $r$ , эклиптической широтой  $\beta$  и эклиптической долготой  $\lambda$  (см. рис.1). *Эклиптической широтой* называется угол между радиус-вектором планеты и плоскостью эклиптики  $Sxy$ . Эклиптическая широта изменяется в пределах от  $0$  до  $90^\circ$  к Северному полюсу эклиптики ( $\Pi$ ) и от  $0$  до  $-90^\circ$  - к ее Южному полюсу ( $\Pi'$ ). *Эклиптической долготой* называется угол, который образует плоскость, проходящая через ось эклиптики  $\Pi\Pi'$  и планету  $P$ , с плоскостью  $Sxz$ . Он изменяется от  $0$  до  $360^\circ$ . Декартовы координаты  $x, y, z$  связаны со сферическими координатами  $r, \beta, \lambda$  соотношениями:

$$\begin{aligned}
 x &= r \cos \lambda \cos \beta \\
 y &= r \sin \lambda \cos \beta \\
 z &= r \sin \beta.
 \end{aligned}$$

Движение планеты по орбите описывается уравнением Кеплера:

$$E - e \sin E = n(t - \tau). \tag{4}$$

Величина  $E$  определяет положение планеты на орбите и называется *эксцентрисической аномалией*. Она имеет простой геометрический смысл: построим окружность радиуса  $a$ , центр которой совпадает с центром эллиптической орбиты  $C$ ; через текущее положение планеты  $P$  проведем перпендикуляр к большой оси эллипса, точку пересечения этого перпендикуляра с окружностью обозначим  $P'$ ; эксцентрисическая аномалия есть угол между радиус-вектором точки  $P'$  относительно  $C$  и направлением на перигелий (рис.4).

Параметр  $n$ , входящий в уравнение (4), выражается через период обращения планеты вокруг Солнца  $T$  (*сидерический период*):

$$n = 2\pi / T. \tag{5}$$

Он равен средней угловой скорости радиус-вектора планеты, в астрономии его называют *средним движением*.

Если ввести обозначение

$$M = n(t - \tau) \tag{6}$$

то уравнение Кеплера примет вид:

$$E - e \sin E = M. \quad (7)$$

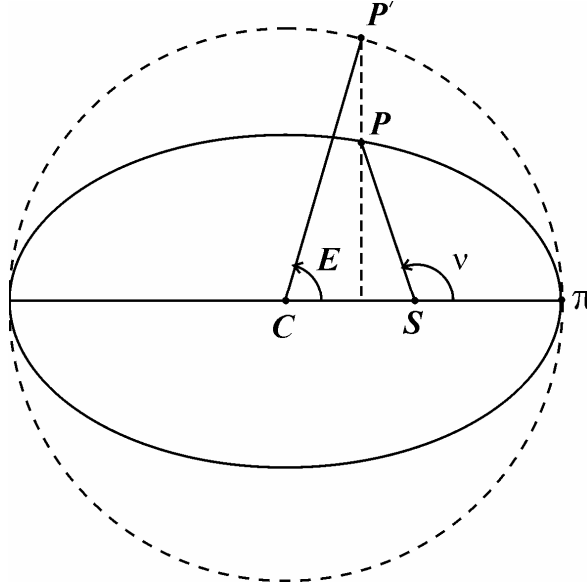


Рис. 4. К определению эксцентрической аномалии

Величину  $M$  называют *средней аномалией*. Если представить себе точку, движущуюся равномерно по окружности радиуса  $a$  и проходящую через перигелий и афелий одновременно с планетой, то положение этой точки будет определяться углом  $M$ , который отсчитывается от направления на перигелий и имеет вершину в центре эллипса.

В качестве начального момента  $t = 0$  в программе SOLAR используется начало астрономической эпохи 2000,0. В этом случае уравнение Кеплера имеет вид:

$$E - e \sin E = M_0 + nt, \quad (8)$$

где угол  $M_0$  определяет положение планеты на орбите в начальный момент времени, он называется *средней долготой в начальную эпоху*.

Расстояние планеты до Солнца  $r$ , истинная аномалия  $v$  и координаты планеты в орбитальной системе  $\xi, \eta$  связаны с эксцентрической аномалией соотношениями:

$$r = a(1 - e \cos E), \quad (9)$$

$$\operatorname{tg} \frac{v}{2} = \sqrt{\frac{1+e}{1-e}} \operatorname{tg} \frac{E}{2}, \quad (10)$$

$$\xi = a \cos E - ae, \quad \eta = b \sin E. \quad (11)$$



Здесь  $b$  - малая полуось эллиптической орбиты:

$$b = a\sqrt{1-e^2} . \quad (12)$$

Учитывая соотношения (1), (11), проекции планеты на оси гелиоцентрической системы координат  $Sxyz$  можно записать в следующем виде:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = a \begin{pmatrix} i_x \\ i_y \\ i_z \end{pmatrix} (\cos E - e) + b \begin{pmatrix} j_x \\ j_y \\ j_z \end{pmatrix} \sin E . \quad (13)$$

Все необходимые для расчета движения планет элементы орбит представлены в Таблице 1.

Таблица 1. Элементы планетных орбит

№	Планета	$a$	$e$	$T$	$i$	$\omega$	$\Omega$	$M_0$
1	Меркурий	0.39 а.е.	0.21	88.0 сут.	$7.00^\circ$	$29^\circ$	$48^\circ$	$252^\circ$
2	Венера	0.72	0.007	224.7 сут.	3.40	54	77	182
3	Земля	1.0	0.017	365.25 сут.	0	103	0	100
4	Марс	1.52	0.09	687.0 сут.	1.85	287	49	335
5	Юпитер	5.20	0.05	11.86 лет	1.30	274	100	32
6	Сатурн	9.55	0.06	29.46 лет	2.48	340	113	50
7	Уран	19.22	0.05	84.01 лет	0.76	99	74	314
8	Нептун	30.11	0.009	164.8 лет	1.77	276	132	304
9	Плутон	39.53	0.25	247.7 лет	17.50	113	110	130

Расстояние от Плутона до Солнца почти в сто раз больше расстояния от Меркурия до Солнца. Поэтому программа SOLAR в каждый момент времени будет отображать на экране или планеты земного типа: Меркурий, Венеру, Землю и Марс, или планеты-гиганты: Юпитер, Сатурн, Уран, Нептун и карликовую планету Плутон.

## 1.2. Решение уравнения Кеплера

Уравнение Кеплера является трансцендентным, поэтому эксцентрическая аномалия  $E$  не может быть представлена в виде элементарной функции времени. Для решения этого уравнения обычно используется метод последовательных приближений.

Уравнение (8) переписывается в виде:

$$E = e \sin E + M_0 + nt. \quad (14)$$

В качестве начального приближения принимается средняя аномалия планеты:

$$E^{(0)}(t) = M_0 + nt, \quad (15)$$

а каждое последующее приближение  $E^{(k+1)}(t)$  находится по предыдущему приближению  $E^{(k)}(t)$  по формуле:

$$E^{(k+1)}(t) = e \sin E^{(k)}(t) + M_0 + nt, \quad k = 0, 1, \dots \quad (16)$$

Модуль производной правой части (14) по неизвестной величине  $E$  меньше единицы:  $|e \cos E| \leq e < 1$ , т.е. достаточное условие сходимости итерационного процесса выполняется, а в силу относительной малости эксцентриситетов орбит планет Солнечной системы, он сходится достаточно быстро.

Итерационный процесс останавливается при выполнении условия:

$$\left| \frac{E^{(k+1)}(t) - E^{(k)}(t)}{E^{(k+1)}(t)} \right| < \varepsilon,$$

где  $\varepsilon$  - максимально допустимая относительная погрешность вычисления решения уравнения, заданная величина.

## Лабораторная работа 1

### ВВОД ЭЛЕМЕНТОВ ОРБИТ И РАСЧЕТ ПАРАМЕТРОВ ДВИЖЕНИЯ ПЛАНЕТ

#### Описание проекта

В первой версии проекта SOLAR вводятся элементы орбит планет  $a$ ,  $e$ ,  $i$ ,  $\omega$ ,  $\Omega$ ,  $M_0$ , период обращения планет вокруг Солнца  $T$ ; вычисляется среднее

движение  $n$ , малые полуоси  $b$  и значения проекций ортов осей  $S\xi$ ,  $S\eta$  орбитальной системы координат на оси гелиоцентрической системы координат.

Форма проекта показана на рис. 5. Она содержит главное меню, имеющее опцию *Солнечная система* и подопцию *Результаты*. При активизации подопции *Результаты* программа открывает окно с результатами расчета среднего движения  $n$ , малых полуосей орбит  $b$  и значений  $ix$ ,  $iy$ ,  $jx$ ,  $ju$  для всех планет.

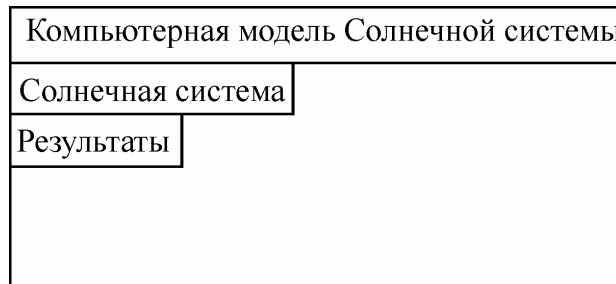


Рис.5. Форма первой версии проекта Solar

В программе принимаются следующие обозначения:

$a$ ,  $b$  - большая и малая полуоси орбиты планеты, в астрономических единицах;

$e$  - эксцентриситет орбиты;

$TP$  - период обращения вокруг Солнца: в рад/сутки – для планет земного типа, в рад/год – для планет-гигантов и Плутона;

$i$  ( $iG$ ) - наклонение орбиты в радианах (градусах);

$omb$  ( $ombG$ ) - долгота восходящего узла  $\Omega$  в радианах (градусах);

$om$  ( $omG$ ) - угловое расстояние перигелия от узла  $\omega$  в радианах (градусах);

$M0$  ( $M0G$ ) - средняя долгота планеты в начальную эпоху в радианах (градусах);

$ix$ ,  $iy$  - проекции орта  $\vec{i}$  на оси  $Sx, Sy$  гелиоцентрической системы координат;

$jx$ ,  $ju$  - проекции орта  $\vec{j}$  на оси  $Sx, Sy$  гелиоцентрической системы координат;

$n$  - среднее движение: в рад/сутки – для планет земного типа, в рад/год – для планет-гигантов и Плутона;

$nameP$  - имя планеты.

### Порядок выполнения работы

1. Описать в секции интерфейсных объявлений величины  $a$ ,  $e$ ,  $TP$ ,  $iG$ ,  $ombG$ ,  $omG$ ,  $M0G$  как типизированные константы – массивы. Объявление типа совместить с заданием элементов массива (использовать Табли-

цу 1). Например, эксцентриситеты орбит можно определить следующим образом:

```
const e: array [1..9] of Double=(0.21, 0.007, 0.017, 0.09, 0.05, 0.06, 0.05,  
                                0.009, 0.25);
```

2. В секции интерфейсных объявлений задать как строковую константу – массив имя планеты:

```
const nameP: array [1..9] of String[8]= ('Меркурий', 'Венера', 'Земля',  
                                           'Марс', 'Юпитер', 'Сатурн', 'Уран', 'Нептун', 'Плутон');
```

3. В секции интерфейсных объявлений описать массивы:

```
var i, omb, om, M0: array [1..9] of Double;
```

4. Включить в обработчик события OnCreate формы расчет следующих параметров:

- а) среднее движение  $n$  - по формуле (5);
- б) малые полуоси орбит  $b$  – по формуле (12);
- в) значения углов  $i$ ,  $omb$ ,  $om$ ,  $M0$  в радианах. Для преобразования градусов в радианы можно воспользоваться методом

```
function DegToRad (Degrees: Extended): Extended;
```

Сослаться на модуль Math, в котором описана эта функция;

- г) значения  $ix$ ,  $iy$ ,  $jx$ ,  $ju$  - по формулам (2), (3).

5. На форме проекта расположить Главное меню (компонент MainMenu), создать опцию *Солнечная система* и подопцию *Результаты*.
6. Подключить к проекту еще одну форму командой

File > New > Form,

установить на ней компонент Мемо, переопределить два свойства этого компонента: Align=alClient (заполнение компонентом всей клиентской области формы), ReadOnly=True (запрет на изменение содержимого окна в процессе работы программы) и составить обработчик события OnClick опции *Результаты*:

```
with Form2.Memo1.Lines do  
  begin  
    Clear;
```

```

for k:=1 to 4 do
    add(nameP[k]+' n='+FloatToStrF ( n[k] ,ffGeneral,5,2)+' рад/сутки');
    . . . . .
end;
Form2.Show;

```

Здесь в качестве образца приведен оператор, отображающий в окне Мемо значение среднего движения для планет земного типа.

В модуле главной формы необходимо сослаться на модуль вспомогательной формы командой:

File > Use Unit.

7. Запустить проект, активизировать опцию *Результаты* и сверить значения  $n$ ,  $b$ ,  $ix$ ,  $iy$ ,  $jx$ ,  $iy$  для Земли и Юпитера с данными, приведенными в Таблице 2.

Таблица 2. Расчетные значения параметров движения Земли и Юпитера

	$n$	$b$	$ix$	$iy$	$jx$	$iy$
Земля	0.017	0.999	-0.225	0.974	-0.974	-0.225
Юпитер	0.053	5.193	0.970	0.242	-0.242	0.970

## Лабораторная работа 2

### ОТОБРАЖЕНИЕ ОРБИТ ПЛАНЕТ

#### Описание проекта

Вторая версия проекта SOLAR представляет на экране ПК проекции орбит планет на плоскость эклиптики  $S_{xy}$ . Она позволяет увидеть причину, по которой Плутон был исключен из состава планет Солнечной системы.

Считается, что плоскость  $S_{xy}$  отображается на поверхность главной формы проекта.

Форма проекта показана на рис.6. В нижней части формы расположена панель, на которой находятся зависимые переключатели: *Земной тип* и *Гиганты*. Подопция главного меню *Показать* выводит на экран орбиты планет земного типа, если активизирована кнопка *Земной тип*, и орбиты планет-гигантов и Плутона, если активизирована кнопка *Гиганты*.



Рис.6. Главное окно второй версии проекта

В программе принимаются обозначения:

$aP$ ,  $bP$  - большая и малая полуоси изображения орбиты планеты, в пикселях;

$colorP$  - цвет планеты и ее орбиты на экране;

$w$ ,  $h$  - ширина и высота окна рисования, которое занимает всю свободную от компонентов клиентскую область главной формы.

Связь между полуосями изображения орбиты планеты задается выражением:

$$bP = aP\sqrt{1-e^2}. \quad (17)$$

Отношение геометрических параметров орбит планет на экране должно соответствовать отношению истинных параметров орбит, т.е. должны выполняться условия:

$$\frac{aP[k]}{a[k]} = \frac{bP[k]}{b[k]}, \quad k = \overline{1, 9}. \quad (18)$$

Наиболее удаленной от Солнца планетой в группе *Земной тип* является Марс. Полуоси орбиты Марса  $aP[4]$ ,  $bP[4]$  необходимо выразить через  $w$  и  $h$  так, чтобы при любом изменении  $w$ ,  $h$  орбита Марса оставалась в пределах окна рисования. После этого из равенств (18) определяются полуоси орбит Меркурия, Венеры и Земли.

Наиболее удаленной от Солнца планетой во второй группе является Плутон. Полуоси орбиты Плутона  $aP[9]$ ,  $bP[9]$  необходимо выразить через  $w$  и  $h$ , после чего полуоси орбит планет-гигантов находятся из условия (18).

Проекция орбиты на плоскость эклиптики задается параметрическими уравнениями:

$$x = ai_x(\cos E - e) + b j_x \sin E, \quad y = ai_y(\cos E - e) + b j_y \sin E. \quad (19)$$

Параметром служит эксцентрическая аномалия  $E$ : когда величина  $E$  пробегает значения от  $0$  до  $2\pi$ , точка с координатами  $x, y$  описывает замкнутую плоскую кривую.

Вывод на форму орбит планет земного типа осуществляет процедура EarthP, а планет-гигантов и Плутона – процедура GiantP. Эти процедуры, в свою очередь, используют процедуру orbit( nP: Integer); которая рисует орбиту одной планеты. Аргумент nP задает номер планеты (первый столбец Таблицы 1).

### Порядок выполнения работы

1. В секции интерфейсных объявлений описать как типизированную константу-массив цвет планеты и орбиты, например:

```
const colorP : array[1..9] of TColor=( clRed, clYellow, clLime, clBlue,  
                                         clRed, clYellow, clLime, clBlue, clSkyBlue);
```

2. Составить процедуру orbit.
3. Установить на форме панель (компонент Panel), переопределить его свойство Align=alBottom.
4. Установить на панели зависимые переключатели: компоненты RadioButton1 (Caption=Земной тип, Checked=True) и RadioButton2 (Caption=Гиганты).
5. Разработать процедуру EarthP, в которой
  - а) вычислить размеры окна рисования  $w$  и  $h$ , координаты Солнца  $xS, yS$  и отобразить Солнце в центре окна;
  - б) определить полуоси орбиты Марса  $aP[4], bP[4]$ ;
  - в) параметры орбит остальных планет земного типа найти из условия (18);
  - г) отобразить на форме орбиты планет земного типа;
  - д) нарисовать ось  $Sx$ . В конце оси поставить метку ' $\gamma$ '. Использовать шрифт MaplePi: Font.Name=MaplePi (он преобразует в ' $\gamma$ ' латинскую букву 'g');
  - е) дать пояснения в окне рисования (рис.7). Использовать шрифт Times New Roman: Font.Name=Times New Roman. Цвет шрифта должен соответствовать цвету орбит.
6. Составить процедуру GiantP.
7. Создать подопцию *Показать* в главном меню проекта, написать для нее обработчик события OnClick.
8. Описать размеры окна рисования  $w, h$ , координаты Солнца  $xS, yS$  и массивы  $aP, bP$  как глобальные переменные:

```
var w, h, xS, yS: Integer;
```

aP, bP: **array**[1..9] of Integer;

- Меркурий
- Венера
- Земля
- Марс

Рис.7. Пояснения в окне рисования

### Лабораторная работа 3

#### РАСЧЕТ И ОТОБРАЖЕНИЕ ПЛАНЕТ НА ОРБИТАХ. ДВИЖЕНИЕ ПЛАНЕТ

##### Описание проекта

Главное окно третьей версии проекта представлено на рис.8. После ввода момента времени  $t$  в строку редактирования и активизации опции *Показать* программа отображает на форме планеты земного типа на орбитах в заданный момент времени  $t$ , если нажата кнопка *Земной тип*, и планеты-гиганты и Плутон, если нажата кнопка *Гиганты*.

Щелчок на опции *Пуск/Стоп* запускает движение планет, начиная с момента времени  $t$ . Следующий щелчок останавливает движение.

Компьютерная модель Солнечной системы			
Солнечная система		Пуск/Стоп	
Результаты			
Показать			
eps	<input type="text"/>	t	<input type="text"/>
kmax	<input type="text"/>	<input checked="" type="radio"/> Земной тип <input type="radio"/> Гиганты	

Рис.8. Форма третьей версии проекта

Движение планеты описывается уравнением Кеплера:

$$E(t) - e \sin E(t) = M_0 + nt. \quad (20)$$



Решение этого уравнения находится методом последовательных приближений: по начальному приближению

$$E^{(0)}(t) = M_0 + nt \quad (21)$$

последующие приближения определяются формулами:

$$E^{(k+1)}(t) = M_0 + nt + e \sin E^{(k)}(t) \quad k = 0, 1 \dots \quad (22)$$

Итерационный процесс прекращается при выполнении одного из условий:

$$\left| \frac{E^{(k+1)}(t) - E^{(k)}(t)}{E^{(k+1)}(t)} \right| < \varepsilon \quad (23)$$

или

$$k = k_{max}. \quad (24)$$

В (23)  $\varepsilon$  - максимально допустимая погрешность вычисления решения уравнения, а  $k_{max}$  - максимальное число итераций; заданные величины.

При выходе из итерационного процесса по условию (23) решение уравнения Кеплера полагается равным  $E^{(k+1)}(t)$ . При выполнении условия (24) решение уравнения Кеплера не определено, в этом случае необходимо остановить итерационный процесс и переопределить значения  $\varepsilon$  и  $k_{max}$ .

Координаты планеты на экране  $x_P, y_P$  связаны с эксцентрической аномалией  $E$  соотношениями:

$$x_P = x_S + a_P i_x (\cos E - e) + b_P j_x \sin E, \quad y_P = y_S - a_P i_y (\cos E - e) + b_P j_y \sin E, \quad (25)$$

где  $a_P, b_P$  - большая и малая полуоси орбиты на экране.

### Порядок выполнения работы

1. Установить на панели три метки (компоненты Label):  $t$ ,  $eps$ ,  $k_{max}$ , рядом с каждой меткой - строку редактирования (компонент Edit) и составить обработчики события OnChange компонентов Edit для передачи в программу значений  $t$ ,  $\varepsilon$  и  $k_{max}$ .
2. В обработчике события OnCreate формы инициализировать значения  $eps$ ,  $k_{max}$  и отобразить эти значения в соответствующих строках редактирования.
3. Для вычисления решения уравнения Кеплера составить функцию

**function** Kepler(t: Extended; e,n,M0,eps:Double; kmax:Integer):Double;

4. Включить в процедуры EarthP, GiantP расчет и отображение планет на орбитах в заданный момент времени  $t$ .
5. Сверить расчетные значения  $E(t)$  в моменты времени  $t=0$  и  $t=10$  с данными, приведенными в Таблице 3.

Таблица 3. Расчетные значения эксцентрической аномалии

Планета	$t=0$	$t=10$
Меркурий	4.21	4.91
Венера	3.18	3.45
Земля	1.76	1.93
Марс	5.81	5.90
Юпитер	0.59	5.83
Сатурн	0.92	3.01
Уран	5.44	6.22
Нептун	5.29	5.68
Плутон	2.43	2.64

6. Определить как константы шаг по времени для планет земного типа (в сутках): StepE и шаг по времени для планет-гигантов (в годах): StepG. Для реализации движения подключить к проекту компонент Timer, расположенный на странице System палитры компонентов и на стадии конструирования формы изменить значение его свойства Enabled на False. Составить обработчик события OnTimer.
7. В главном меню проекта создать опцию *Пуск/Стоп*. В обработчик события OnClick этой опции вставить оператор:

Timer1.Enabled:=not Timer1.Enabled;

для запуска и остановки движения планет.

8. В режиме анимации представить на экране «мерцающие» звезды, число которых, положение и цвет должны определяться в программе случайным образом. Использовать свойство

Pixels[X, Y :Integer] : TColor

объекта Canvas, позволяющее устанавливать и читать цвет каждой точки на холсте (X, Y – координаты точки).

## Лабораторная работа 4

### ВВОД И ОТОБРАЖЕНИЕ ДАТЫ. ОПЦИЯ *ФАЙЛ*

#### Описание проекта

Форма проекта показана на рис.9. При активизации подопции *Дата* открывается окно с предложением ввести краткую дату (для планет земного типа) или год (для планет-гигантов). После ввода даты команда *Показать* представляет положение планет в заданный момент времени, а команда *Пуск/Стоп* запускает движение, начиная с этого момента времени. В процессе движения текущее время в виде краткой даты (для планет земного типа) и года (для планет-гигантов) отображается в строке редактирования с меткой  $t$ .

Компьютерная модель Солнечной системы		
Файл	Солнечная система	Пуск/Стоп
Сохранить как	Результаты	
Копировать	Дата	
Удалить	Показать	
Вставить		
Выход		
<div> <div> <math>eps</math> <input type="text"/> <math>kmax</math> <input type="text"/> </div> <div> <math>t</math> <input type="text"/> </div> <div> <input checked="" type="radio"/> Земной тип  <input type="radio"/> Гиганты         </div> </div>		

Рис.9. Главное окно четвертой версии проекта

Подопция *Сохранить как* позволяет сохранить изображение, представленное в окне рисования, в виде BMP - файла. Подопция *Копировать* копирует изображение в буфер межпрограммного обмена. Подопция *Удалить* очищает окно рисования. Подопция *Вставить* вставляет изображение из буфера. Команда *Выход* завершает работу проекта.

Для отображения времени можно воспользоваться переменной типа `TDateTime`. Такая переменная представляет собой вещественное число, целая часть которого определяет дату, т.е. количество суток, прошедших с 30 декабря 1899 года. Функция

`StrToDate (St: String): TDateTime`

преобразует символы строки `St` в дату. Строка `St` должна содержать три целых числа, разделенных точкой, т.е. имеет формат:

‘дд.мм.гггг’ (26)

Первое число определяет правильный день, второе – правильный месяц, а третье – год. Функция

DateToStr (Value: TDateTime): String

преобразует дату Value в строку формата (26).

Для улучшения качества анимации в этой версии проекта используется переменная типа TBitmap (битовая карта). Такая переменная обладает свойством Canvas, которое позволяет создавать на ней изображение с помощью графических инструментов Pen, Brush, Font и методов класса TCanvas. Для представления движения при каждом наступлении события OnTimer изображение, соответствующее текущему моменту времени, сначала выводится на поверхность битовой карты, а затем копируется на поверхность формы.

### Порядок выполнения работы

1. Очистить обработчик события OnChange компонента Edit с меткой *t*. Включить в процедуру Timer1Timer вывод текущего времени в окно этого компонента. Для планет земного типа текущую дату возвращает функция:

DateToStr (StrToDate('01.01.2000')+t).

2. Создать подопцию *Дата* для ввода краткой даты для планет земного типа и года – для планет-гигантов. Воспользоваться методом InputBox: например, для планет земного типа:

St:=InputBox ('Дата', 'Введите дату в виде дд.мм.гггг', '01.01.2000');  
t:= StrToDate (St) – StrToDate ('01.01.2000') ;

для планет-гигантов:

St:=InputBox('Дата', 'Введите год', '2000');  
t:=                    ;

Здесь St – переменная типа String.

3. Описать как глобальную переменную битовую карту:

**var** BitKadr: TBitmap;

В обработчике события OnCreate формы создать эту переменную и определить ее размеры:

```

Bitkadr:=TBitmap.Create;
Bitkadr.Width := Screen. Width;   Bitkadr.Height := Screen. Height;

```

В процедурах EarthP, GiantP изменить заголовок оператора присоединения:

```

      Bitkadr
with Form1.Canvas do
. . .

```

После вызова этих процедур скопировать изображение с поверхности битовой карты на поверхность формы:

```

Canvas.CopyRect (Rect (0, 0, w, h), Bitkadr.Canvas, Rect(0, 0, w, h));

```

В обработчике события OnDestroy формы удалить битовую карту из оперативной памяти:

```

Bitkadr.Free;

```

4. Создать опцию *Файл* и подопции *Сохранить как*, *Копировать*, *Удалить*, *Вставить* и *Выход*. В качестве источника изображения при выполнении команд *Сохранить как*, *Копировать* и приемника изображения при выполнении команды *Вставить* использовать переменную Bitkadr.
5. Предусмотреть в программе удаление отпечатков окон, которые открывают команды *Сохранить как* и *Дата*. Использовать для этого переменную BitKadr.

## Лабораторная работа 5

### ОКНО ПЛАНЕТЫ

#### Описание проекта

Главная форма проекта показана на рис.10. При активизации опции главного меню *Планеты* появляется окно, в котором представлены все планеты Солнечной системы и Солнце (рис.11), причем их относительные размеры на экране соответствуют их истинным относительным размерам.

Компьютерная модель Солнечной системы			
Файл	Солнечная система	Пуск/Стоп	Планеты
<i>eps</i> <input type="text"/> <i>kmax</i> <input type="text"/>	<i>t</i> <input type="text"/>	<input checked="" type="radio"/> Земной тип <input type="radio"/> Гиганты	

Рис.10. Главное окно пятой версии проекта

В программе принимаются следующие обозначения:

$rP$  ( $r$ ) - массив линейных размеров планет в  $m$  (пикселях);

$rSm$  ( $rS$ ) - линейный размер Солнца в  $m$  (пикселях);

$xC$  - массив горизонтальных координат центров планет в окне *Планеты*;

$xS$ ,  $yS$  - координаты центра Солнца в окне *Планеты* (центр расположен вне окна рисования);

$dw$  - расстояние между планетами.

Значения элементов массива  $rP$  приведены в Таблице 4. Линейный размер Солнца  $rSm = 6.96 \cdot 10^8 m$ .

Радиусы планет и Солнца на экране должны удовлетворять соотношениям:

$$\frac{r[k]}{rP[k]} = \frac{rS}{rSm}, \quad k = \overline{1,9}. \quad (27)$$

Если задать значение  $r[1]$  для наименьшей из планет – Меркурия, то радиусы остальных планет и Солнца определяются равенствами (27). Расстояние между планетами и горизонтальные координаты их центров вычисляются по формулам:

$$dw = \frac{w - 2 \sum_{i=1}^9 r[i]}{10} \quad (28)$$

$$xC[1] = dw + r[1], \quad xC[k+1] = xC[k] + r[k] + r[k+1] + dw, \quad k = \overline{1,8}. \quad (29)$$

Здесь  $w$  - ширина окна рисования.

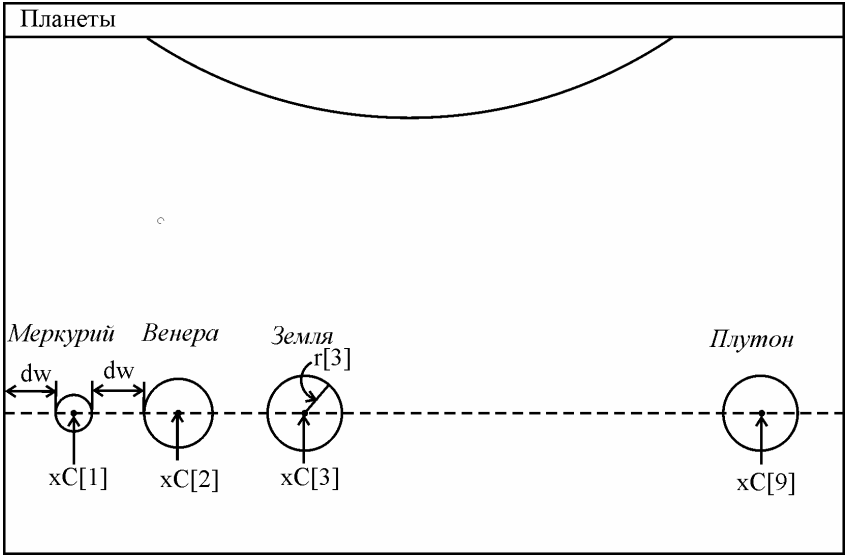


Рис.11. Окно Form3

Таблица 4. Линейные размеры планет (в м)

Меркурий	Венера	Земля	Марс	Юпитер	Сатурн	Уран	Нептун	Плуто́н
$2.4 \cdot 10^6$	$6.2 \cdot 10^6$	$6.4 \cdot 10^6$	$3.4 \cdot 10^6$	$7.1 \cdot 10^7$	$6.0 \cdot 10^7$	$2.4 \cdot 10^7$	$2.2 \cdot 10^7$	$7.2 \cdot 10^6$

**Порядок выполнения работы**

1. Подключить к проекту новую форму – Form3 и максимально увеличить ее ширину. Определить свойства формы: Caption=Планеты, Border-Style=bsDialog (запрет на изменение размеров формы в процессе работы программы) и Position=poScreenCenter (окно в момент появления располагается в центре экрана и имеет такие же размеры, как на этапе конструирования).
2. В обработчике события OnPaint для Form3 задать rP (как константу – массив), rSm (как константу), определить r, dw, xC, xS, yS и отобразить на поверхности формы планеты и сегмент Солнца. Рядом с каждой планетой вывести ее название.
3. Создать опцию Планеты в главном меню проекта и составить обработчик события OnClick этой опции.

4. В модуле Form3 сослаться на модуль главной формы, а в модуле главной формы – на модуль Form3.
5. Переменные *r*, *xC*, *dw* описать как глобальные в секции реализации.

## Лабораторная работа 6

### ИНФОРМАЦИЯ О ПЛАНЕТАХ

#### Описание проекта

Шестая версия проекта после щелчка на одной из планет в окне *Планеты* открывает новую форму, которая содержит фотографию выделенной планеты и краткую информацию о ней.

#### Порядок выполнения работы

1. Скопировать в папку проекта файлы 1.bmp – 9.bmp, содержащие изображения планет (имя файла совпадает с номером планеты в Таблице 1).
2. Добавить к проекту новую форму Form4, уменьшить ее размеры, запретить изменение размеров формы в процессе выполнения программы.
3. Установить на Form4 многострочный редактор (компонент Memo), прижать его к правой границе формы (Align=alRight), изменить значение свойства ReadOnly на True.
4. Разместить на Form4 компонент Image, заполнить им всю клиентскую область формы.
5. Создать обработчик события OnMouseDown для Form3. Эта процедура должна определять номер numberP выделенной планеты и открывать окно Form4. Использовать событие

OnMouseDown (Sender: TObject; Button: TMouseButton;  
Shift: TShiftState; X, Y: Integer),

которое наступает при нажатии кнопки мыши на форме. При этом параметр Button содержит нажатую кнопку (mbLeft – левая, mbRight – правая, mbMiddle – средняя), а параметры X, Y - координаты указателя мыши. Переменную numberP: Integer описать как глобальную в секции интерфейсных объявлений.

6. Для вывода на экран выделенной планеты вставить в обработчик события OnActivate для Form4 код:



```

w:=Image1.Width; h:=Image1.Height;
FName:=IntToStr(numberP)+'.bmp';
if FileExists(FName) then
    begin{1}
        Planet:=TBitmap.Create;
        Planet.LoadFromFile(FName);
        wI:=Planet.Width; hI:=Planet.Height
    end{1}
else begin{2}
    with Image1.Canvas do
        begin{3}
            Brush.Color:=Color;
            FillRect(Rect(0,0,w,h))
        end{3};
        ShowMessage('Файл'+FName+'отсутствует'); Exit
    end{2} ;
with Image1.Canvas do
    begin{4}
        Brush.Color:=Planet.Canvas.Pixels[3,3];
        FillRect(Rect(0,0,w,h));
        StretchDraw(Rect( , , , ), Planet);
        Planet.Free
    end{4} ;

```

Здесь изображение планеты сначала копируется методом LoadFromFile из файла в битовую карту Planet, а затем переносится на поверхность компонента Image1 методом StretchDraw. Аргументы этого метода определить самостоятельно так, чтобы фотография планеты занимала максимально возможную площадь на Form4, но при этом сохранялись относительные размеры изображения. Цвет заполнения окна Image1 должен совпадать с цветом фона изображения.

7. Описать локальные переменные:

```

var Planet: TBitmap;  FName: String;
    w, h, wI, hI : Integer;

```

8. В окно компонента Мемо вывести значения элементов орбиты выделенной планеты (использовать Таблицу 1).
9. В заголовок окна Form4 вынести имя выделенной планеты.
10. В модуле Form4 сослаться на модули главной формы и Form3.

## Лабораторная работа 7

### ГРАФИКИ

#### Описание проекта

Седьмая версия проекта SOLAR вычисляет и представляет в графическом виде зависимость от времени скорости  $V$  выделенной планеты и ее расстояния от Солнца  $r$ . Главное меню проекта содержит опцию *Графики* и две подопции:  $r(t)$  и  $V(t)$ . Щелчок мыши на подопции  $r(t)$  ( $V(t)$ ) открывает окно, в котором при движении планет вычерчивается график функции  $r(t)$  ( $V(t)$ ).

Расстояние планеты от Солнца задается выражением:

$$r = a(1 - e \cos E), \quad (30)$$

где  $E$  - эксцентрическая аномалия. Оно минимально в перигелии:  $r_{\min} = a(1 - e)$  и максимально в афелии:  $r_{\max} = a(1 + e)$ .

Скорость планеты можно найти из интеграла энергии:

$$V^2 - \frac{2\mu}{r} = h, \quad (31)$$

где  $\mu = \gamma(m_S + m_P) \approx \gamma m_S \approx 1,33 \cdot 10^{20} \text{ м}^3 / \text{с}^2$ ,  $\gamma$  - гравитационная постоянная,  $m_S$  - масса Солнца, а  $m_P$  - масса планеты. Учитывая (30) и значение постоянной энергии для эллиптического движения:  $h = -\mu/a$ , из (31) получаем:

$$V = \sqrt{\frac{\mu}{a} \frac{1 + e \cos E}{1 - e \cos E}}. \quad (32)$$

В перигелии  $E = 0$  и скорость планеты максимальна:

$$V_{\max} = \sqrt{\frac{\mu}{a} \frac{1 + e}{1 - e}}. \quad (33)$$

В афелии  $E = \pi$  и скорость минимальна:

$$V_{\min} = \sqrt{\frac{\mu}{a} \frac{1 - e}{1 + e}}. \quad (34)$$

Поскольку функции  $r(t)$ ,  $V(t)$  периодические, в качестве интервала изменения времени принимается отрезок  $[t, t + T]$ , где  $t$  - момент активизации окна графика, а  $T$  - период обращения планеты вокруг Солнца. В качестве интервалов изменения  $r(t)$  и  $V(t)$  используются отрезки  $[r_{\min}, r_{\max}]$  и  $[V_{\min}, V_{\max}]$ , соответственно.

Программа представляет график в виде ломаной линии: при каждом наступлении события OnTimer вычерчивается новое звено ломаной, соединяющее точку, отвечающую предыдущему моменту времени  $t_{pre}$ , с точкой, отвечающей текущему моменту времени  $t$  (рис.12).

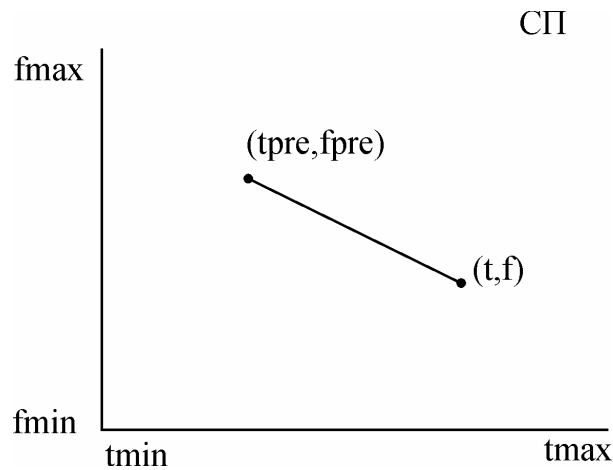


Рис.12. Система координат пользователя

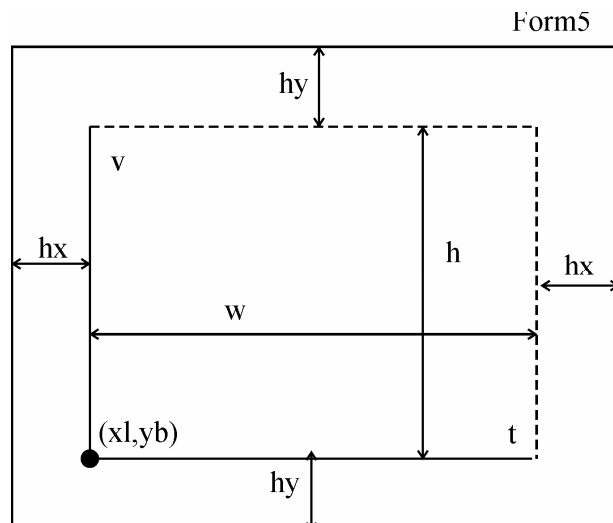
Координаты точки в системе координат пользователя  $(t, f)$  связаны с координатами точки в графической системе окна рисования  $(x, y)$  соотношениями:

$$x = xl + m_x(t - t_{\min}), \quad y = yb - m_y(f - f_{\min}), \quad (35)$$

где

$$m_x = \frac{w}{t_{\max} - t_{\min}}, \quad m_y = \frac{h}{f_{\max} - f_{\min}} - \quad (36)$$

масштабирующие множители, одинаковые для всех точек графика;  $w$ ,  $h$  - размеры графика на экране;  $xl$ ,  $yb$  - координаты левой нижней точки графика (рис.13).

Рис.13. Окно графика  $V(t)$ 

Для параметров  $\mu$ ,  $m_x$ ,  $m_y$ ,  $t_{\min}$ ,  $t_{\max}$ ,  $V_{\min}$ ,  $V_{\max}$ ,  $t_{pre}$ ,  $V_{pre}$  в программе приняты обозначения:  $\mu$ ,  $m_x$ ,  $m_y$ ,  $t_{\min}$ ,  $t_{\max}$ ,  $V_{\min}$ ,  $V_{\max}$ ,  $t_{pre}$ ,  $V_{pre}$ .

### Порядок выполнения работы

1. Определить как константу величину  $\mu$ :

$$\mu = 1.33 \times 10^{20};$$

2. В обработчик события OnCreate главной формы вставить оператор:

$$\text{numberP} := 4;$$

т.е. по умолчанию выделенной планетой считается Марс.

3. Подключить к проекту новую форму – Form5, уменьшить ее размеры и определить заголовок: *График  $V(t)$* . В обработчике события OnPaint формы задать поля графика  $hx$ ,  $hy$  (выразить через размеры клиентской области формы), вычислить размеры графика  $w$ ,  $h$ , координаты левой нижней точки  $x_l$ ,  $y_b$  и масштабирующие множители  $m_x$ ,  $m_y$ . Залить поверхность формы цветом фона, нарисовать оси и вывести метки на осях.
4. В главном меню проекта создать опцию *Графики* и две подопции:  $r(t)$  и  $V(t)$ .
5. В обработчике события OnClick подопции  $V(t)$  выполнить следующие действия:
  - а) вычислить значения  $t_{\min}$ ,  $t_{\max}$ ,  $V_{\min}$ ,  $V_{\max}$ ;
  - б) определить координаты первой точки графика в системе пользователя  $t_{pre}$ ,  $V_{pre}$ ;

- с) открыть окно Form5.
6. Вставить в процедуру Timer1Timer операторы, отображающие в окне Form5 текущее звено графика  $V(t)$ .
7. Переменные  $t_{pre}$ ,  $V_{pre}$  описать как глобальные в секции реализации. Переменные  $x_l$ ,  $y_b$ ,  $m_x$ ,  $m_y$ ,  $t_{min}$ ,  $t_{max}$ ,  $V_{min}$ ,  $V_{max}$  описать как глобальные в секции интерфейсных объявлений.
8. Самостоятельно разработать программную реализацию подопции  $r(t)$ .

## Лабораторная работа 8

### СОЕДИНЕНИЕ (ПРОТИВОСТОЯНИЕ) ПЛАНЕТ

#### Описание проекта

Восьмая версия проекта SOLAR осуществляет поиск планетных конфигураций типа соединения (противостояния) планеты с Солнцем.

Планеты, орбиты которых расположены внутри орбиты Земли, называются *внутренними* или *нижними*. Планеты, орбиты которых лежат за пределами орбиты Земли, называются *внешними* или *верхними*. Внутренними планетами являются Меркурий и Венера, остальные планеты – внешние.

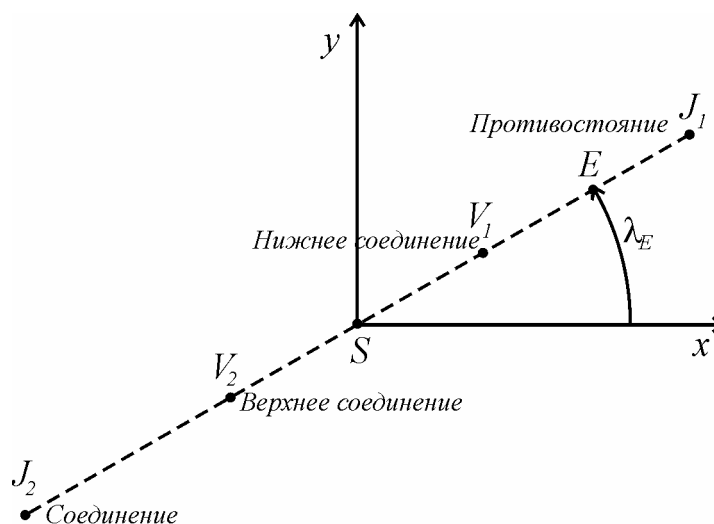


Рис. 14. Планетные конфигурации ( $E$  - Земля,  $S$  - Солнце,  $V$  - внутренняя планета,  $J$  - внешняя планета)

Если планета проходит через прямую Земля – Солнце, говорят о *соединении* планеты с Солнцем в том случае, когда направление на планету относительно Земли совпадает с направлением на Солнце (конфигурации  $ESV_2$ ,  $ESJ_2$ ,  $EV_1S$ ), и о *противостоянии* (*оппозиции*), когда направление на планету противоположно направлению на Солнце (конфигурация  $SEJ_1$ ) (рис.14). Внутренняя планета может находиться в *верхнем соединении* (конфигурация  $ESV_2$ ) и в *нижнем соединении* (конфигурация  $EV_1S$ ).

При соединении (противостоянии) с Солнцем светила, орбита которого принадлежит плоскости эклиптики, его эклиптическая долгота совпадает с эклиптической долготой Земли или отличается от нее на  $180^0$ . Эклиптическая долгота связана с координатами светила  $x, y$  равенством:

$$\operatorname{tg} \lambda = \frac{y}{x}. \quad (37)$$

В момент противостояния с Солнцем внешней планеты и нижнего соединения внутренней планеты эклиптическая долгота планеты  $\lambda_P$  совпадает с эклиптической долготой Земли  $\lambda_E$ :

$$\lambda_P - \lambda_E = 0. \quad (38)$$

Во время соединения с Солнцем внешней планеты и верхнего соединения внутренней планеты эклиптические долготы планеты и Земли различаются на угол  $180^0$ :

$$\lambda_P - \lambda_E = \pm 180^0. \quad (39)$$

В действительности плоскости орбит не совпадают с плоскостью эклиптики, но близки к ней, поэтому условия соединения (противостояния) (38), (39) выполняются только приближенно.

Соединение (противостояние)
Глубина поиска
Шаг поиска
Поиск
Конфигурация

Рис.15. Опция меню *Соединение (противостояние)*

Главное меню восьмой версии проекта SOLAR содержит опцию *Соединение (Противостояние)* и подопции *Глубина поиска*, *Шаг поиска*, *Поиск* и *Конфигурация* (рис.15). Команда *Поиск* находит ближайшее по времени к текущему моменту  $t$  соединение (противостояние) выделенной планеты с Солнцем, определяет и выдает вид планетной конфигурации (нижнее или верхнее соединение для внутренней планеты и соединение или противостояние для внешней планеты) и дату. Команда *Конфигурация* представляет на экране положение планет на орбитах в найденный момент времени. Подопция *Глубина поиска* предназначена для ввода промежутка времени  $\tau$ , для которого проводится поиск: расположение планет рассматривается во временном интервале  $(t, t + \tau)$ , если  $\tau > 0$ , и в интервале  $(t + \tau, t)$ , если  $\tau < 0$ . Подопция *Шаг поиска* позволяет задать погрешность  $\Delta t$ , с которой будет получено время соединения (противостояния). Программа находит и отображает на экране конфигурации планет земного типа.

### Порядок выполнения работы

1. В главном меню проекта создать опцию *Соединение (Противостояние)* и подопции *Глубина поиска*, *Шаг поиска*, *Поиск* и *Конфигурация*.
2. Написать обработчики события OnClick подопции *Глубина поиска* для ввода значения  $\tau$  (в годах) и подопции *Шаг поиска* для ввода значения  $\Delta t$  (в сутках). Воспользоваться функцией InputBox.
3. Реализовать программно поиск соединения (противостояния) с Солнцем планет земного типа. Эклиптическую долготу  $\lambda$  находить по координатам планеты в системе пользователя методом

**function** ArcTan2(Y, X: Extended): Extended,

который вычисляет арктангенс  $Y/X$  и возвращает угол в правильном квадранте (функция ArcTan не учитывает квадрант). Сослаться на модуль Math, в котором описан метод ArcTan2. Для определения планетных конфигураций воспользоваться равенствами (38), (39). Условием соединения (противостояния) считать изменение знака  $\sin(\lambda_P - \lambda_E)$ . При этом для противостояния внешней планеты и нижнего соединения внутренней планеты выполняется неравенство  $\cos(\lambda_P - \lambda_E) > 0$ , а для соединения внешней планеты и верхнего соединения внутренней планеты - неравенство  $\cos(\lambda_P - \lambda_E) < 0$ .

4. Разработать подопцию *Конфигурация*. Отобразить время конфигурации в виде краткой даты в строке редактирования Edit1.
5. В процедуре FormCreate инициализировать значения  $\tau = 50$  и  $\Delta t = 0.01$  (они будут считаться принятыми по умолчанию).

## Лабораторная работа 9

### ЗАВЕРШЕНИЕ РАБОТЫ НАД ПРОЕКТОМ

#### Описание проекта

Главное меню последней версии проекта SOLAR содержит опцию *О программе*, которая открывает окно с информацией о проекте и её разработчике. Под строкой главного меню находится инструментальная панель, кнопки которой дублируют все скрытые команды (подопции) меню. Кнопки инструментальной панели снабжены подсказками: короткими (ярлычками), появляющимися при задержке указателя мыши около кнопки, и длинными или развернутыми, которые отображаются на панели, расположенной в нижней части главной формы SOLAR.

#### Порядок выполнения работы

1. Подключить к проекту компонент `ImageList` (хранилище изображений), расположенный на странице Win32 палитры компонентов. Поместить в хранилище столько изображений, сколько подопций в меню. Для загрузки изображения двойным щелчком мыши на объекте `ImageList` открыть встроенный редактор, в окне редактора нажать кнопку `Add` и выбрать любой BMP – файл из каталога `Images > Buttons`. Файлы этого каталога содержат два изображения: для кнопки в активном состоянии и для кнопки в пассивном состоянии. Второе изображение удалить из хранилища, выделив его и щелкнув на кнопке `Delete`.
2. Добавить к проекту компонент `ToolBar` (инструментальная панель), расположенный на странице Win32 палитры компонентов. Установить в свойство `Images` этого компонента значение `ImageList1`.
3. Разместить на инструментальной панели кнопки `ToolButton` (командой `New Button` контекстного меню компонента `ImageList`). Кнопки, соответствующие разным опциям меню, разделить сепараторами (команда `New Separator` контекстного меню). В конец инструментальной панели вынести кнопку, связанную с командой *Выход*.
4. Присвоить свойству `Images` главного меню значение `ImageList1`.
5. Разместить на форме компонент `Panel` (панель `Panel2`) и прижать его к расположенной в нижней части формы панели `Panel1`. Подключить к проекту компонент `ApplicationEvents` (события приложения), расположенный на странице `Additional` палитры компонентов; для отображения развернутых подсказок на `Panel2` вставить в обработчик события `OnHint` оператор:

`Panel2.Caption:=Application.Hint;`

а для того, чтобы программа показывала ярлычки, свойству `ShowHint` главной формы присвоить значение `True`.



6. Для каждой подопции главного меню проекта задать два свойства: ImageIndex (номер изображения в хранилище) и Hint (подсказки). В свойство Hint необходимо поместить текстовую строку, состоящую из двух частей, разделенных символом «|»: первая часть определяет содержимое ярлычка, а вторая - развернутой подсказки.
7. Для каждой кнопки инструментальной панели определить свойство MenuItem – раздел меню, дублируемый кнопкой. Чтобы кнопки «щелкали», в свойство Style поместить значение tbsCheck (кнопка - переключатель).
8. Подключить к проекту заготовку для формы AboutBox из репозитория Delphi (команда File > New > Other, страница Forms), заполнить форму AboutBox информацией о программе и её разработчике и сохранить в папке проекта. Создать опцию меню *О программе* и составить обработчик события OnClick этой опции.
9. Сделать работу всех пунктов меню согласованной: в каждый момент времени работы программы пользователю должны быть доступны только те команды, которые могут быть выполнены в этот момент без ошибок.

### 1.3. Задания по проекту SOLAR

- Реализовать в программе поиск соединения двух выбранных пользователем планет. У к а з а н и е. *Соединением* называется положение двух тел Солнечной системы на одной линии с Землей. В такой конфигурации эклиптические долготы тел относительно Земли равны и они кажутся расположенными на небе почти в одном и том же месте.
- Отображать на панели расстояние до Солнца и истинную аномалию движущихся планет.
- Включить в проект расчет моментов максимальной элонгации выделенной внутренней планеты и восточной (западной) квадратуры выделенной внешней планеты. У к а з а н и е. *Элонгацией* планеты называется угол между радиус-вектором планеты и радиус-вектором Солнца относительно Земли. Когда элонгация планеты равна  $90^\circ$ , говорят, что планета *находится в квадратуре*.
- В процедуре orbit использовать разный стиль для отображения части орбиты, расположенной над плоскостью эклиптики (сплошная линия) и под ней (пунктирная линия).
- Пояснения в окне рисования разместить вблизи планет.
- Предусмотреть в программе возможность одновременного представления в разных окнах движения группы планет земного типа и планет-гигантов.
- В качестве начального момента времени использовать в программе дату своего рождения. У к а з а н и е. В уравнении Кеплера среднюю аномалию на 01.01.2000  $M_0$  заменить величиной  $M_0 - n\Delta t$ , где  $\Delta t$  - промежу-

ток времени, прошедший с момента рождения до 01.01.2000. Учесть новый отсчет времени при вводе и отображении даты.

- Добавить к проекту окно, которое должно содержать расстояния между движущимися планетами активной группы.
- Вставить в программу поиск соединения (противостояния) с Солнцем планет-гигантов.
- Снабдить Form2 контекстным меню, содержащим пункты: *Сохранить*, *Копировать*, *Удалить*, *Вставить* и *Заккрыть*. Первая команда должна сохранять результаты расчета параметров движения в текстовом файле, вторая – копировать информацию в буфер, третья – очищать окно с результатами расчетов, четвертая – вставлять данные из буфера, а последняя – закрывать Form2.
- Предусмотреть в программе возможность представления движения внешних планет.
- Добавить к проекту окно, в которое будут выводиться эклиптические координаты движущихся планет (эклиптическая широта  $\beta$  и эклиптическая долгота  $\lambda$ ).
- Отображать на экране отдельно движение внутренних и внешних планет.
- Включить в проект расчет планетных конфигураций, при которых две или более планет проходят через линию Солнце-Земля.
- Разбить планеты на две группы следующим образом: Меркурий - Юпитер, Сатурн – Плутон.

## 2. ПРОЕКТ НОНMANN

### 2.1. Перевод космического аппарата с одной круговой орбиты на другую - задача Гомана

Перелет космического аппарата с одной планеты на другую условно можно разделить на три этапа: 1) полет в области притяжения первой планеты, 2) движение в поле тяготения Солнца до границ области притяжения второй планеты, 3) полет в области притяжения второй планеты. Предположим, что орбиты планет являются круговыми и компланарными, и, кроме того, можно пренебречь силами притяжения планет. Рассмотрим движение по гелиоцентрической эллиптической траектории, которая в перигеуме касается круговой орбиты одной планеты, а в апоцентре – круговой орбиты другой планеты. Такая орбита называется *орбитой* или *эллипсом Гомана* - по имени немецкого ученого Вальтера Гомана (Walter Hohmann, 1880-1943), который в начале XX века занимался проблемой межпланетных полетов. Она обеспечивает движение между орбитами с наименьшими затратами энергии.

Обозначим через  $r_1$ ,  $r_2$  радиусы орбит планет, причем для определенности будем считать, что  $r_1 < r_2$  (рис.16). Для осуществления перелета космическому аппарату необходимо сообщить два импульса: первый (в точке  $\pi$ ) переводит аппарат с меньшей круговой орбиты на эллиптическую орбиту, а второй (в точке  $A$ ) - с эллиптической орбиты на бóльшую круговую орбиту. Эти импульсы прилагаются по касательной к орбите путем кратковременного включения двигателя, направленного в сторону, противоположную движению.

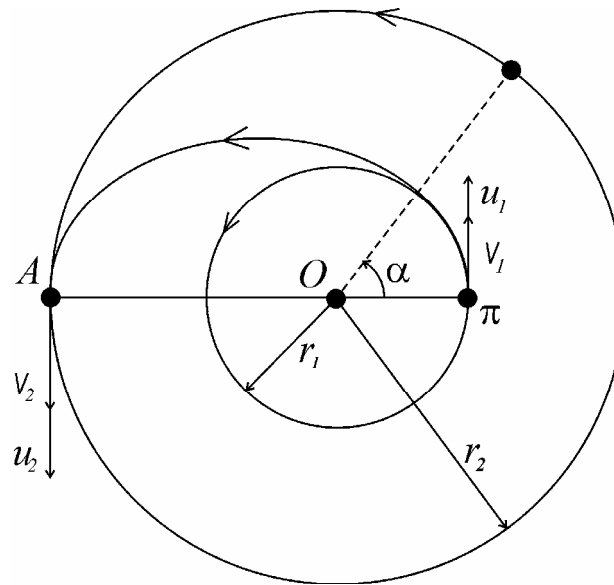


Рис.16. К постановке задачи Гомана

Скорость движения по круговой орбите радиуса  $r_1$  задается выражением

$$V_1 = \sqrt{\frac{\mu}{r_1}}, \quad (40)$$

где

$$\mu = \gamma m_O, \quad (41)$$

а  $m_O$  - масса Солнца.

Большая полуось эллиптической орбиты связана со скоростью в перигелии соотношением:

$$(a =) \frac{r_1 + r_2}{2} = \frac{\mu}{2\mu/r_1 - (u_1 + V_1)^2}. \quad (42)$$

Здесь  $u_1$  - дополнительная скорость в точке  $\pi$ .

Из закона сохранения момента количества движения аппарата относительно центра  $O$  следует:

$$V_2 r_2 = (V_1 + u_1) r_1, \quad (43)$$

где  $V_2$  - скорость, с которой аппарат приходит в апоцентр орбиты Гомана.

Дополнительная скорость  $u_2$  в точке  $A$  подбирается так, чтобы результирующая скорость аппарата совпала со скоростью движения по круговой орбите радиуса  $r_2$ :

$$V_2 + u_2 = \sqrt{\frac{\mu}{r_2}}. \quad (44)$$

Решая систему уравнений (40), (42)-(44), для  $u_1$ ,  $u_2$  получаем:

$$u_1 = \sqrt{\frac{2\mu r_2}{r_1(r_1 + r_2)}} - \sqrt{\frac{\mu}{r_1}}, \quad (45)$$

$$u_2 = \sqrt{\frac{\mu}{r_2}} - \sqrt{\frac{2\mu r_1}{r_2(r_1 + r_2)}}. \quad (46)$$

Суммарное приращение скорости аппарата  $u_1 + u_2$  называется *энергетическими затратами* перелета, эта величина измеряется в км/с.

Время запуска аппарата должно быть определено таким образом, чтобы к моменту его прохождения через точку  $A$  в эту же точку попала вторая планета. Пусть космический аппарат стартует в момент времени  $t = t_0$ , а через точку  $A$  проходит в момент времени  $t = t_1$ . Движение аппарата описывается уравнением Кеплера:

$$E - e \sin E = n(t - t_0), \quad (47)$$

где  $e$  - эксцентриситет орбиты:

$$e = \frac{r_2 - r_1}{r_2 + r_1},$$

$n$  - среднее движение:

$$n = \left( \frac{2}{r_1 + r_2} \right)^{3/2} \mu^{1/2}.$$

Из (47) следует:

$$(E(t_1) =) \pi = n(t_1 - t_0),$$

откуда получаем:

$$t_1 - t_0 = \frac{\pi}{n} = \frac{\pi}{\mu^{1/2}} \left( \frac{r_1 + r_2}{2} \right)^{3/2}. \quad (48)$$

Обозначим через  $\alpha$  угол, который составляет в момент  $t = t_0$  радиус-вектор второй планеты с радиус-вектором первой планеты, он называется *углом начальной конфигурации*. За время  $t_1 - t_0$  вторая планета, двигаясь со скоростью  $\sqrt{\mu/r_2}$ , должна описать дугу длиной  $r_2(\pi - \alpha)$  (см. рис.16), следовательно

$$r_2(\pi - \alpha) = \sqrt{\frac{\mu}{r_2}}(t_1 - t_0).$$

Отсюда определяем  $\alpha$  :

$$\alpha = \pi \left[ 1 - \left( \frac{r_1 + r_2}{2r_2} \right)^{3/2} \right]. \quad (49)$$

Модуль скорости аппарата в любой момент времени  $t$  можно найти из интеграла энергии:

$$V^2 - \frac{2\mu}{r} = h. \quad (50)$$

Здесь  $r = a(1 - e \cos E)$  - расстояние до точки  $O$ ;  $h$  - постоянная энергии, которая определяется из начальных условий:

$$h = (V_1 + u_1)^2 - \frac{2\mu}{r_1}.$$

Из (50) находим:

$$V(t) = \sqrt{h + \frac{2\mu}{a[1 - e \cos E(t)]}}. \quad (51)$$

Движение планет задается уравнениями:

$$\varphi_1(t) = \frac{\mu^{1/2}}{r_1^{3/2}} t, \quad \varphi_2(t) = \frac{\mu^{1/2}}{r_2^{3/2}} t + \alpha,$$

где  $\varphi_i$  - угол, который образует радиус-вектор  $i$ -той планеты относительно точки  $O$  с направлением на перицентр орбиты Гомана.

Полученные соотношения справедливы и в том случае, когда движение происходит в поле тяготения планеты, а космический аппарат совершает перелет между круговыми орбитами вокруг этой планеты. При этом в выражении (41)  $m_O$  - масса планеты.

Формулы (45)-(51) описывают также перелет с внешней орбиты на внутреннюю орбиту, если через  $r_1, r_2$ , как и ранее, обозначить радиус орбиты запуска и орбиты стыковки, соответственно ( $r_1 > r_2$ ). В этом случае значения  $u_1, u_2$  отрицательны и дополнительные скорости направлены противоположно скоростям  $V_1, V_2$ .

## 2.2. Описание проекта

Программа НОНМАНН рассчитывает и отображает на экране ПК перелет космического аппарата по гомановской траектории между произвольно выбранными пользователем планетами Солнечной системы в поле притяжения Солнца, а также между произвольно заданными круговыми орбитами в поле тяготения любой планеты. Форма проекта представлена на рис.17. Главное меню содержит опции: *Планеты*, *Гомановская орбита*, *Пуск/Стоп*, *Графики* и *О программе*. Под строкой меню находится инструментальная панель, кнопки которой дублируют скрытые команды (подопции) меню. В нижней части формы установлены две панели. На верхней панели имеется кнопка *Солнце*: если она включена, рассматривается движение в поле тяготения Солнца, в этом случае зависимые переключатели *Первая орбита*, *Вторая орбита* и подопции меню *Планеты* позволяют выбрать планеты, между которыми осуществляется перелет. Если кнопка *Солнце* выключена, движение происходит в поле тяготения одной из планет, которая задается подопцией меню *Планеты*, а строки редактирования с метками  $r1/rP$ ,  $r2/rP$  предназначены для ввода отношений радиусов орбит к радиусу планеты ( $r1/rP, r2/rP > 1$ ).


Компьютерная модель задачи Гомана				
Планеты	Гоман.орбита	Пуск/Стоп	Графики	О программе
Меркурий	Расчет Результаты Показать Сохранить		E(t) V(t)	
Венера				
Земля				
Марс				
Юпитер				
Сатурн				
Уран				
Нептун				
Плутон				
Выход				
<div> <math>t</math> <input type="text"/> <div> <input type="radio"/> Первая орбита           <input checked="" type="radio"/> Вторая орбита         </div> <div>  Солнце         </div> <div> <math>r1/rP</math> <input type="text"/>  <math>r2/rP</math> <input type="text"/> </div> </div>				

Рис.17. Главное окно проекта НОНMANN

По команде *Расчет* программа вычисляет основные параметры движения, а подопция *Результаты* открывает окно с результатами счета. Команда *Показать* отображает конфигурацию системы в момент времени, который отсчитывается с момента запуска аппарата и вводится в строку с меткой  $t$  на верхней панели, подопция *Сохранить* сохраняет изображение в виде BMP – файла. Пункт меню *Пуск/Стоп* запускает и останавливает движение, подпункты  $E(t)$ ,  $V(t)$  открывают окна, в которых в процессе движения отображаются графики зависимости эксцентрической аномалии и модуля скорости аппарата от времени. При движении на нижней панели работает индикатор прогресса, а во время остановки та же панель служит для отображения развернутых подсказок кнопок инструментальной панели.

### 2.3. План лабораторных работ

1. Расчет основных параметров перелета с Земли на Марс.
2. Расчет и изображение планет и космического аппарата в заданный момент времени.
3. Движение планет и аппарата.
4. Перелеты между планетами в поле тяготения Солнца.

5. График  $E(t)$ .
6. График  $V(t)$ .
7. Движение в поле тяготения планеты.
8. Перелеты с внешней круговой орбиты на внутреннюю.
9. Завершение работы над проектом.

### 3. ПРОЕКТ GRAVITY

#### 3.1 Движение в поле тяготения планеты

Рассмотрим движение тела в поле тяготения планеты, пренебрегая действием на тело Солнца и других планет. Введем систему координат, связанную с планетой. Предположим, что планета имеет сферическую форму, тогда движущееся тело можно моделировать материальной точкой, находящейся в центральном ньютоновском поле сил.

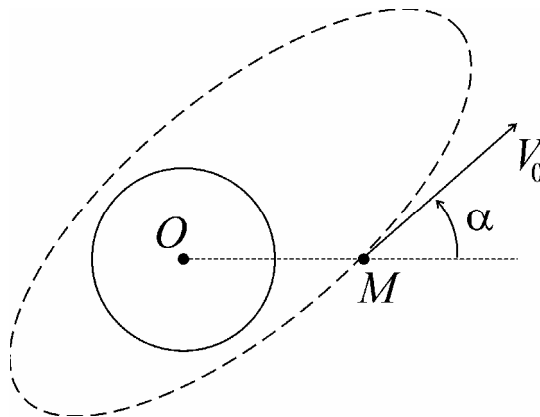


Рис. 18. К постановке задачи о движении тела в поле тяготения планеты

Пусть в начальный момент времени  $t = t_0$  точка занимает положение  $M$  на высоте  $H$  от поверхности планеты и имеет скорость  $\vec{V}_0$ , которая составляет угол  $\alpha$  с радиус-вектором точки относительно центра планеты  $O$  (рис.18). Обозначим через  $r_P$  радиус планеты, а через  $r_0$  - расстояние точки до центра планеты при  $t = t_0$ :  $r_0 = r_P + H$ . Если задано начальное положение  $M$ , то характер орбиты определяется величиной начальной скорости. Если  $V_0 < V_P$ , где  $V_P = \sqrt{2\mu/r_0}$  - параболическая скорость, точка движется по эллипсу (эллиптическое движение); при  $V_0 = V_P$  - по параболе (параболическое движение), а при  $V_0 > V_P$  - по гиперболе



(гиперболическое движение). Эллиптическое движение является финитным: оно происходит в ограниченной области пространства, в этом случае полная энергия точки отрицательна и кинетической энергии не хватает для преодоления гравитационной силы притягивающего центра. Параболическое и гиперболическое движения – инфинитны: полная энергия неотрицательна, движущаяся точка преодолевает силу притяжения планеты и уходит на бесконечность. Если постоянная площадей

$$c = r_0 V_0 \sin \alpha \quad (52)$$

равна нулю, т.е. обращается в нуль кинетический момент точки относительно центра  $O$ , то движение вырождается в прямолинейное.

Программа GRAVITY моделирует эллиптическое движение точки.

Большая полуось и эксцентриситет эллиптической орбиты определяются начальными условиями:

$$a = \frac{\mu}{2\mu/r_0 - V_0^2}, \quad (53)$$

$$e = \sqrt{1 + h \frac{c^2}{\mu^2}}, \quad (54)$$

где  $\mu = \gamma m_P$ ,  $m_P$  - масса планеты;  $h$  - постоянная энергии:

$$h = V_0^2 - \frac{2\mu}{r_0}. \quad (55)$$

Из (53), (55) следует, что большая полуось  $a$  и постоянная  $h$  являются функциями длины радиус-вектора и квадрата скорости, т.е. они зависят от величины скорости, но не зависят от ее направления. Угол  $\alpha$  входит в постоянную площадей  $c$  и влияет на эксцентриситет орбиты (54).

Движение точки описывается уравнением Кеплера (4):

$$E - e \sin E = n(t - \tau),$$

где  $n$  - среднее движение:

$$n = \frac{\mu^{1/2}}{a^{3/2}}.$$

Из соотношения (9) эксцентрическая аномалия в начальный момент времени выражается через расстояние точки до центра планеты:

$$E_0 = \arccos \frac{a - r_0}{ae},$$

а из (10) по  $E_0$  находится истинная аномалия при  $t = t_0$ :

$$v_0 = 2 \operatorname{arctg} \left( \sqrt{\frac{1+e}{1-e}} \operatorname{tg} \frac{E_0}{2} \right).$$

Модуль скорости точки в произвольный момент времени определяется из интеграла энергии:

$$V(t) = \sqrt{h + \frac{2\mu}{a[1 - e \cos E(t)]}}.$$

Если в некоторый момент времени  $t = t_l$  выполняется равенство

$$(r =) a(1 - e \cos E) = r_p, \quad (56)$$

то точка, описав дугу эллипса, упадет на поверхность планеты. Условие (56) выполняется в том случае, когда имеет место неравенство:

$$(|\cos E| =) \frac{1}{e} \left| 1 - \frac{r_p}{a} \right| < 1. \quad (57)$$

Время движения по такой траектории задается выражением:

$$t_l - t_0 = \frac{1}{n} [E_l - E_0 - e(\sin E_l - \sin E_0)],$$

где  $E_l$  - эксцентрическая аномалия в момент встречи с планетой, она является решением уравнения (56).

Если неравенство (57) не выполняется, точка совершает периодическое движение по эллиптической орбите с периодом:

$$T = 2\pi \frac{a^{3/2}}{\mu^{1/2}}.$$

### 3.2. Описание проекта

Программа GRAVITY рассчитывает и отображает на экране ПК эллиптическое движение тела в поле тяготения любой выбранной пользователем планеты

Солнечной системы. Форма проекта представлена на рис.19. Главное меню содержит опции: *Планеты*, *Начальные условия*, *Орбита*, *Пуск/Стоп*, *Графики* и *О программе*. Под строкой меню находится инструментальная панель, кнопки которой дублируют скрытые команды (подопции) меню. В нижней части формы установлены две панели. На верхней панели имеется три строки редактирования, с помощью которых можно управлять положением и относительным размером изображения планеты на экране.

Движение в поле тяготения планеты					
Планеты	Нач.условия	Орбита	Пуск/Стоп	Графики	О программе
Меркурий	Высота	Расчет		E(t)	
Венера	Скорость	Результаты		V(t)	
Земля	Угол	Сохранить			
Марс					
Юпитер					
Сатурн					
Уран					
Нептун					
Плутон				Время	
Выход				Показать	
				Копировать	
				Удалить	
				Вставить	
				Сохранить как	
<div> <div>t <input type="text"/></div> <div>rP/W <input type="text"/></div> <div>xP/W <input type="text"/></div> <div>yP/W <input type="text"/></div> </div>					

Рис.19. Главное окно проекта GRAVITY

Первая опция меню позволяет выбрать планету, в поле тяготения которой происходит движение. Вторая опция предназначена для ввода начальных условий: высоты  $H$ , величины скорости  $V_0$  и угла  $\alpha$ . Команды опции *Орбита* вычисляют основные параметры движения, выводят результаты расчета и сохраняют их в виде текстового файла. Пункт меню *Пуск/Стоп* запускает и останавливает движение, подпункты  $E(t)$ ,  $V(t)$  открывают окна, в которых в процессе движения отображаются графики зависимости эксцентрической аномалии и модуля скорости тела от времени. На нижней панели расположен индикатор прогресса, состояние которого синхронизировано с движением точки. Во время остановки та же панель служит для отображения развернутых подсказок кнопок инструментальной панели. В строку редактирования с меткой  $t$  выводится время, отсчитываемое от начального момента  $t_0$ .

Программа имеет контекстное меню, которое появляется на форме по щелчку правой кнопки мыши. Оно состоит из пунктов: *Время*, *Показать*, *Копировать*, *Удалить*, *Вставить* и *Сохранить как*. Команды этого меню позволяют за-

дать время, отсчитываемое от начала движения, представить на экране тело на орбите вблизи планеты в этот момент времени, скопировать изображение в буфер, очистить окно рисования, вставить изображение из буфера и сохранить его в виде BMP-файла.

### 3.3. План лабораторных работ

1. Расчет основных параметров движения тела в поле тяготения Земли.
2. Отображение планеты и орбиты.
3. Расчет и изображение тела на орбите в заданный момент времени.
4. Движение при произвольных начальных условиях.
5. Движение в поле тяготения планет Солнечной системы.
6. График  $E(t)$ .
7. График  $V(t)$ .
8. Разработка контекстного меню.
9. Завершение работы над проектом.

### СПИСОК ИСПОЛЬЗОВАННОЙ И РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Маркеев А.П. Теоретическая механика: Учебное пособие для университетов. – Ижевск: НИЦ "РХД", 1999.
2. Справочное руководство по небесной механике и астродинамике/ Под ред. Дубошина Г.Н. – М.:Наука, 1971.
3. Александров Ю.В. Небесна механіка. – Харків: ХНУ імені В.Н.Каразіна, 2004.
4. Рой А. Движение по орбитам. – М.: Мир, 1981.
5. Белецкий В.В. Очерки о движении космических тел. – М.: Наука, 1977.
6. Левантовский В.И. Механика космического полета в элементарном изложении. – М.: Наука, 1980.
7. Fix John D. Astronomy: journey to the cosmic frontier. – McGraw-Hill, 1999.
8. Бобровский С.И. Delphi 7. Учебный курс. – СПб.: Питер, 2005.
9. Тюкачев Н., Свиридов Ю. Delphi 5. Создание мультимедийных приложений. – СПб.: Питер. – 2001.
10. Фаронов В. Delphi – 2005: язык, среда, разработка приложений. – СПб.: Питер. – 2005.

### Средства отладки в Delphi

Перед запуском проекта Delphi в отладочном режиме, его необходимо откомпилировать. Предварительно надо настроить компилятор для максимального контроля ошибок. Для этого нужно открыть страницу **Compiler** меню **Project > Options** и выполнить следующие действия:

1. На панели **Code generation** (Генерация машинного кода) выключить опцию **Optimization** (Оптимизация);
2. На панели **Runtime errors** (Ошибки времени выполнения) включить опции: **Range Checking** (Контроль выхода индекса за границы массива), **I/O Checking** (Контроль ошибок ввода-вывода) и **Overflow checking** (Контроль переполнения при целочисленных операциях);
3. На панели **Debugging** (Отладка) включить опции: **Debug information** (Информация об отладке), **Local symbols** (Просмотр значений локальных переменных), **Reference info** (Просмотр структуры кода), **Assertions** (Включение процедуры **Assert** в машинный код) и **Use Debug DCUs** (использование отладочных версий стандартных модулей библиотеки компонентов **VCL**).

**Выполнение по шагам.** Для выполнения программы по шагам в ней надо определить *точку прерывания*, дойдя до которой программа приостановит свою работу и временно передаст управление системе Delphi. Точку прерывания можно задать для любой строки исходного текста, которая содержит исполняемый оператор. Такие строки помечены в левом поле окна редактора кода круглыми маркерами, которые появляются после успешно выполненной компиляции. Точка прерывания определяется щелчком мыши на маркере, при этом соответствующая строка выделяется цветом; повторный щелчок мыши удаляет ее. Если после задания точки прерывания запустить программу, она дойдет до выделенной строки и остановится. Во время остановки программы можно проверить значения переменных, наводя на них указатель мыши.

Выполнение программы можно продолжить по шагам. Для этого используется команда **Run > Step Over** (Запуск > Перешагнуть), клавиша **F8** или кнопка **Step Over** инструментальной панели системы. Для входа в подпрограммы, вложенные в модуль, применяется команда **Run > Trace into** (Запуск > Войти внутрь), клавиша **F7** или кнопка **Trace into** инструментальной панели системы. Выйти из подпрограммы, пропустив группу операторов, можно командой **Run > Run Until Return** (Запуск > Выполнить до выхода), в результате управление передается на оператор, следующий за вызовом данной подпрограммы.

Прекратить работу проекта можно командой **Run > Program Reset** (Запуск > Сброс программы).

Чтобы пропустить выполнение группы операторов, можно использовать команду **Run > Run to Cursor** (Запуск > Выполнение до курсора) или нажать кла-

вишу **F4**. Предварительно надо установить курсор в строку, с которой требуется продолжить выполнение по шагам.

Продолжить выполнение программы можно командой **Run** (Запуск) или клавишей **F9**.

**Просмотр значений.** Система Delphi позволяет отслеживать значения переменных в режиме отладки в специальном окне. Добавление новых переменных в это окно осуществляется командой **Run > Add Watch** (Запуск > Добавить слежение) или клавишей **Ctrl+F5**. В открывающемся при этом окне **Watch Properties** (Свойства просмотра) имя переменной вводится в поле **Expression** (Выражение). Можно следить за значением не одной переменной, а целого выражения, которое в этом случае необходимо задать в указанном поле.

В нижней части окна **Watch Properties** имеется ряд переключателей, с помощью которых можно указать тип результирующего выражения. По умолчанию отображается значение, имеющее тот же тип, что и переменная – переключатель **Default** (По умолчанию).

В поле **Repeat count** (Число элементов) указывается, сколько элементов массива следует отображать в окне просмотра.

В поле **Digits** (Разрядность) задается число значащих цифр после запятой, которые будут выводиться для нецелых чисел.

Опция **Allow Function Calls** (Разрешить вызов функций) позволяет использование в выражениях функций.

Чтобы отредактировать выражение, ранее внесенное в список просмотра, надо дважды щелкнуть на нем. Удаление выражения производится нажатием клавиши **DELETE**. В контекстном меню окна просмотра имеются опции **Delete All Watches** (Удалить все элементы), **Disable All Watches** (Отключить все элементы), **Enable All Watches** (Включить все элементы), позволяющие удалить, временно отключить или включить все элементы, соответственно.

Для того чтобы окно просмотра находилось поверх других окон, нужно включить опцию **Stay on Top** (Поверх других окон) контекстного меню окна.

**Прерывание по условию.** Чтобы прервать работу цикла при определенном значении параметра цикла (например, при  $i=3$ ), к точке прерывания, определенной внутри цикла, надо добавить условие, при котором эта точка сработает. Для этого необходимо поместить курсор внутри цикла и вызвать команду **Run > Add Breakpoint > Source Breakpoint** (Запуск > Добавить точку прерывания > Исходная точка прерывания), на экране появится диалоговое окно **Add Source Breakpoint** (Добавление исходной точки прерывания). В поле **Condition** (Условие) надо ввести выражение  $i=3$  и щелкнуть на кнопке **OK**.

## ОГЛАВЛЕНИЕ

	ПРЕДИСЛОВИЕ	3
1.	ПРОЕКТ SOLAR	4
	1.1. Невозмущенное движение планет	
	1.2. Решение уравнения Кеплера	10
	Лабораторная работа 1. Ввод элементов орбит и расчет параметров движения планет	
	Лабораторная работа 2. Отображение орбит планет	13
	Лабораторная работа 3. Расчет и отображение планет на орбитах. Движение планет	16
	Лабораторная работа 4. Ввод и отображение даты. Опция <i>Файл</i>	19
	Лабораторная работа 5. Окно <i>Планеты</i>	21
	Лабораторная работа 6. Информация о планетах	24
	Лабораторная работа 7. Графики	26
	Лабораторная работа 8. Соединение (противостояние) планет	29
	Лабораторная работа 9. Завершение работы над проектом	32
	1.3. Задания по проекту SOLAR	33
2.	ПРОЕКТ HOHMANN	34
	2.1. Перевод космического аппарата с одной круговой орбиты на другую - задача Гомана	
	2.2. Описание проекта	38
	2.3. План лабораторных работ	39
3.	ПРОЕКТ GRAVITY	40
	3.1. Движение в поле тяготения планеты	
	3.2. Описание проекта	42
	3.3. План лабораторных работ	44
	Список использованной и рекомендуемой литературы	
	Приложение. Средства отладки в Delphi	45

Навчальне видання

**Попова Лілія Миколаївна**

**Комп'ютерне моделювання задач небесної механіки**

Російською мовою

Редактор І.Ю.Агаркова

Коректор А.І.Меліхова

Комп'ютерна верстка Л.М.Попова

Розробка дизайну макета обкладинки І.М.Дончик

Підписано до друку . Формат 60х84/16.

Папір офсетний. Друк ризографічний.

Ум.-друк. арк. . Обл.-вид. арк. . Наклад 50 прим.

Ціна договірна

61077, Харків, майдан Свободи, 4,  
Харківський національний університет імені В.Н.Каразіна,  
Видавництво ХНУ імені В.Н.Каразіна

Надруковано ФОП «Петрова І.В.»  
61144, Харків, 144, вул. Гв. Широнівців, 79-в, к.137.  
Тел. 362-01-52

Свідоцтво про державну реєстрацію ВОО №948011 від 03.01.03.